Rapids Simulator

What is RAPIDS?

- An event-driven simulator for distributed real-time systems
- Simulator itself also distributed, runs on a heterogeneous cluster of workstations

Issues in Real-Time Systems

- Timeliness
- Recovery from failure
- Fault-tolerance

Why RAPIDS?

- Assist in the development and testing of recovery algorithms

- Analyze other hardware and software components of real-time systems

Simulator Structure

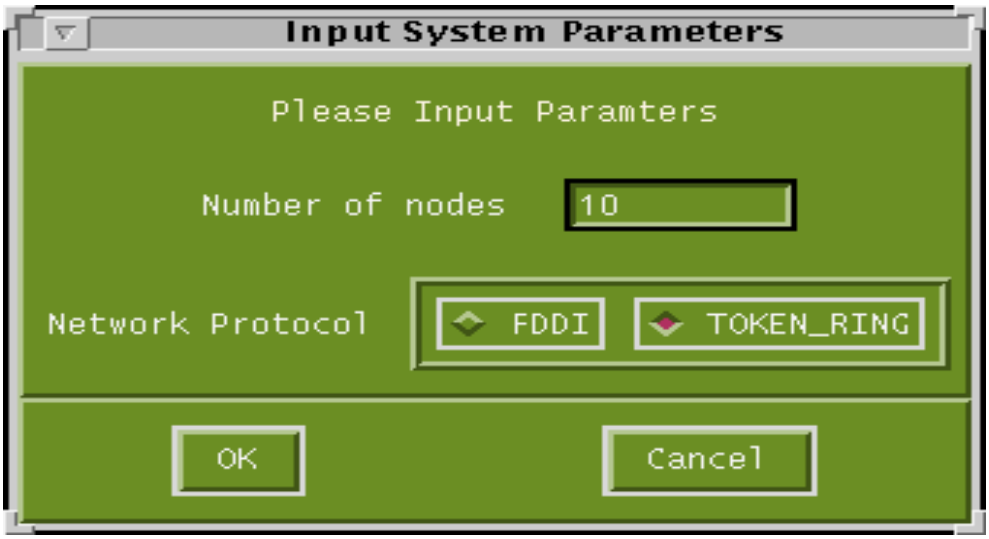
- **Physical machines**
- **PVM message passing interface**
- **PVM processes**
- **Simulated system**

Simulated System Components

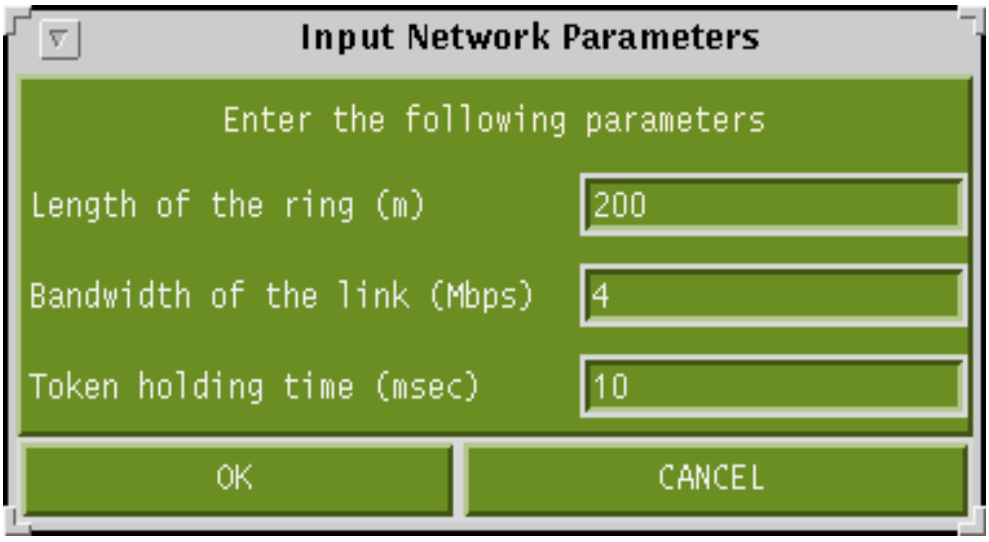
- **Console - The link between the GUI and the simulator**
- **Clock - Ensures causality, uses the breathing time buckets technique**
- **Network - Simulates connection network between nodes.**
- **Node(s)**

Node Components

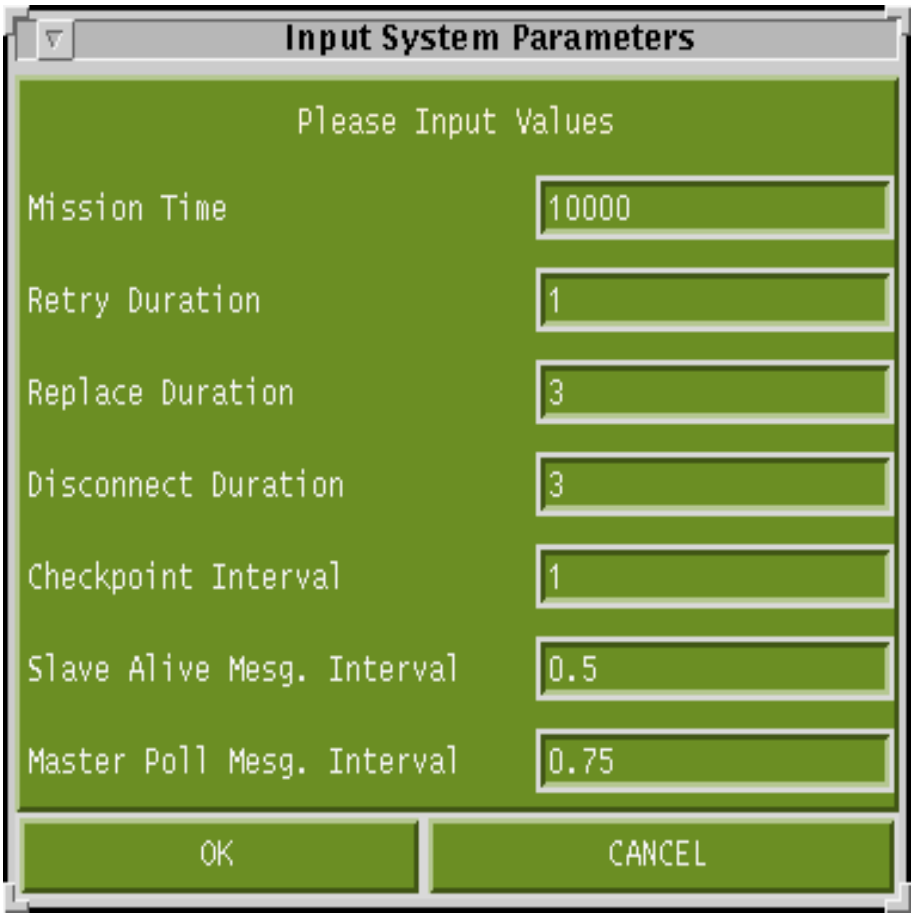
- **Tasks: Synthetic and real**
- **Scheduling algorithms**
- **Allocation algorithms**
- **Checkpointing**
- **Recovery actions**



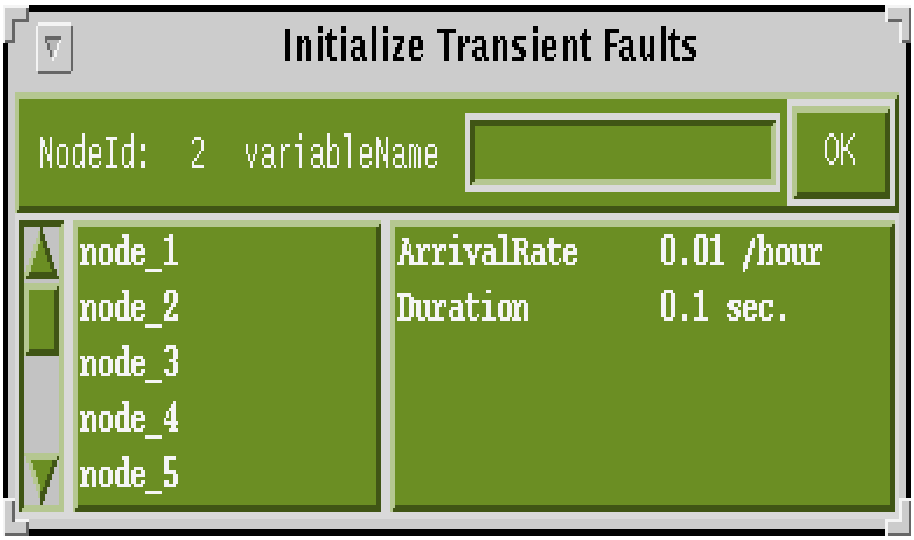
Choice of network protocol and number of nodes



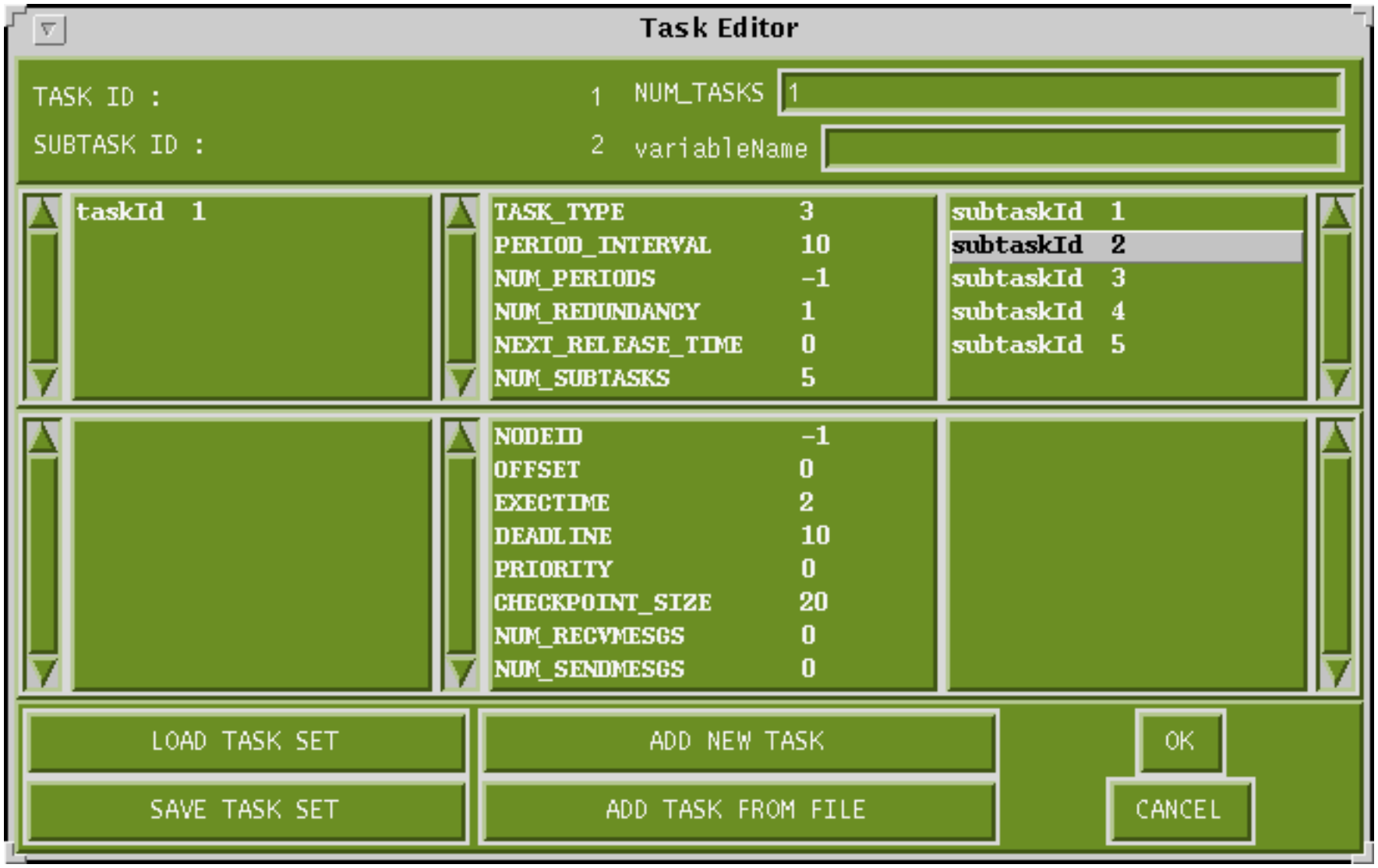
Choice of network parameters



Choice of various system and recovery parameters



Choice of fault rates


Task Editor
TASK ID :
1
NUM_TASKS
1
SUBTASK ID :
2
variableName
taskId 1
TASK_TYPE 3
PERIOD_INTERVAL 10
NUM_PERIODS -1
NUM_REDUNDANCY 1
NEXT_RELEASE_TIME 0
NUM_SUBTASKS 5
subtaskId 1
subtaskId 2
subtaskId 3
subtaskId 4
subtaskId 5
NODEID -1
OFFSET 0
EXECTIME 2
DEADLINE 10
PRIORITY 0
CHECKPOINT_SIZE 20
NUM_RECVMESGS 0
NUM_SENDMESGS 0
LOAD TASK SET
SAVE TASK SET
ADD NEW TASK
ADD TASK FROM FILE
OK
CANCEL

Recovery Algorithms

- **Static, in which the user specifies ahead of time what fixed recovery action is to be taken**

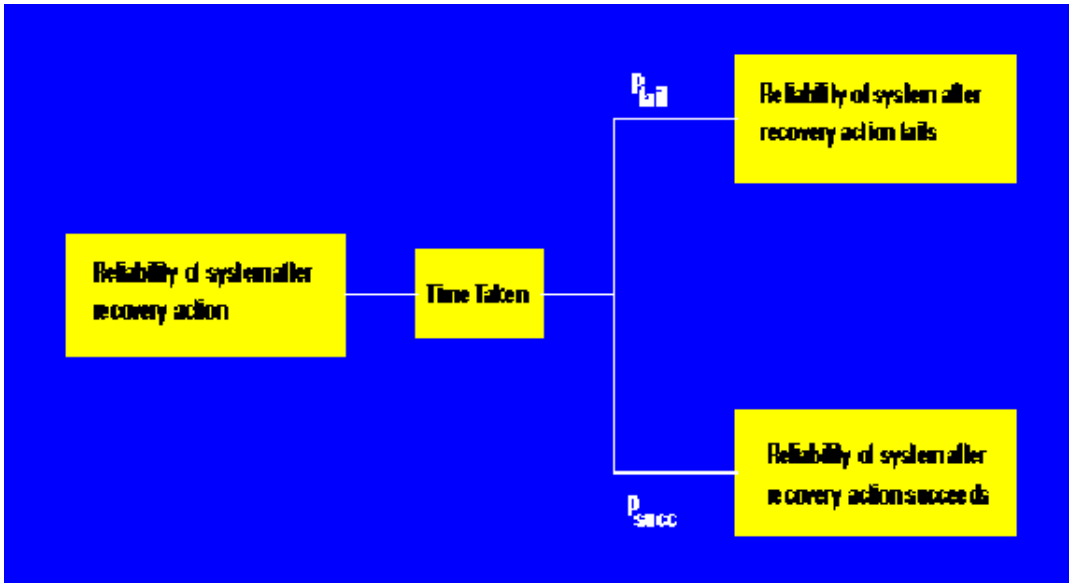
- **Our RAMP algorithm, a dynamic offline algorithm**

Recovery Actions

- Retry the task from its last checkpoint
- Replace the faulty node by a spare (if available)
- Disconnect the faulty node and redistribute its tasks over the remaining nodes

RAMP Algorithm

- RAMP generates efficient recovery policies for a given system depending on:
  - ⋆ Operating environment
  - ⋆ Workload
  - ⋆ System failure characteristics
  - ⋆ Remaining mission time


Reliability of system after recovery action
Time Taken
p_{fail}
Reliability of system after recovery action fails
p_{succ}
Reliability of system after recovery action succeeds

Why a Benchmark?

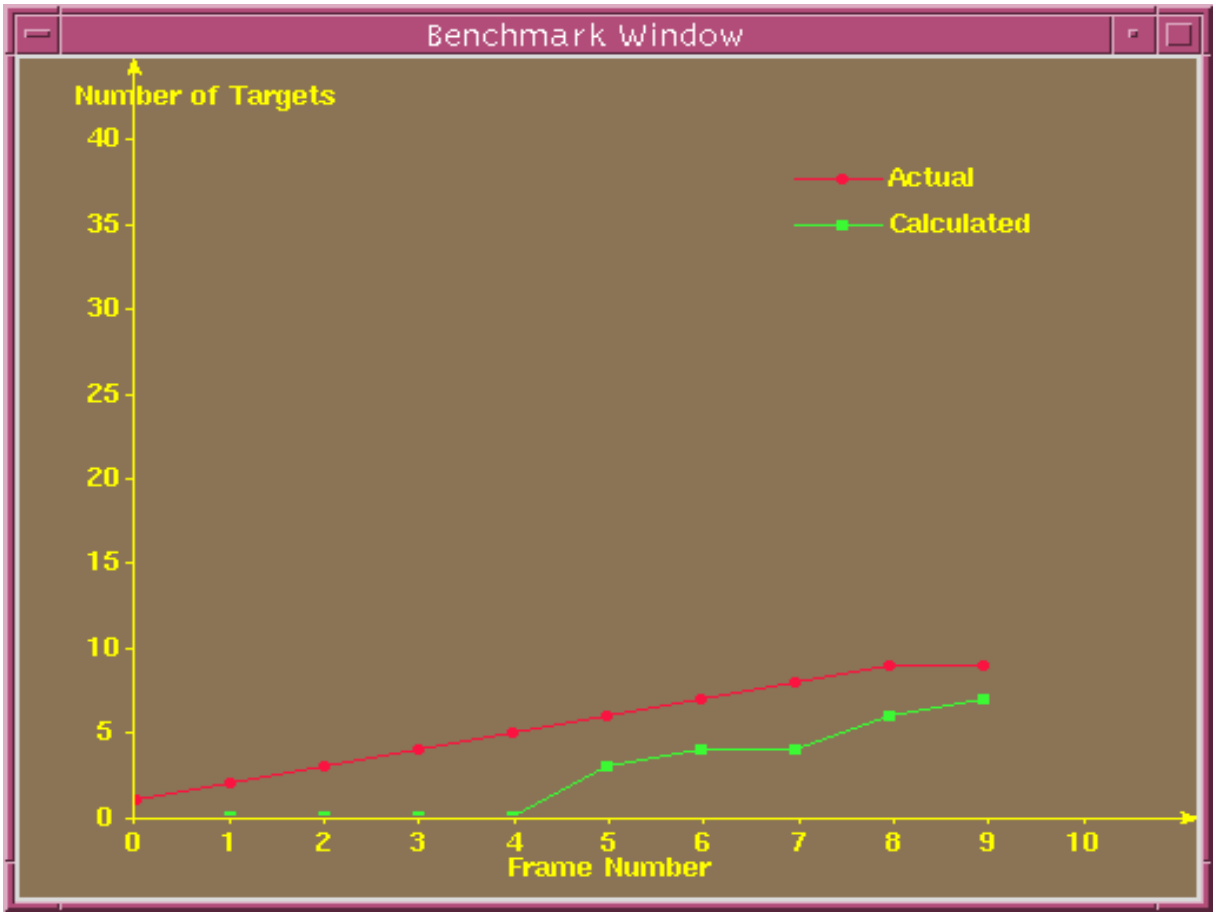
- Analyze the performance impact of the implemented algorithms and policies

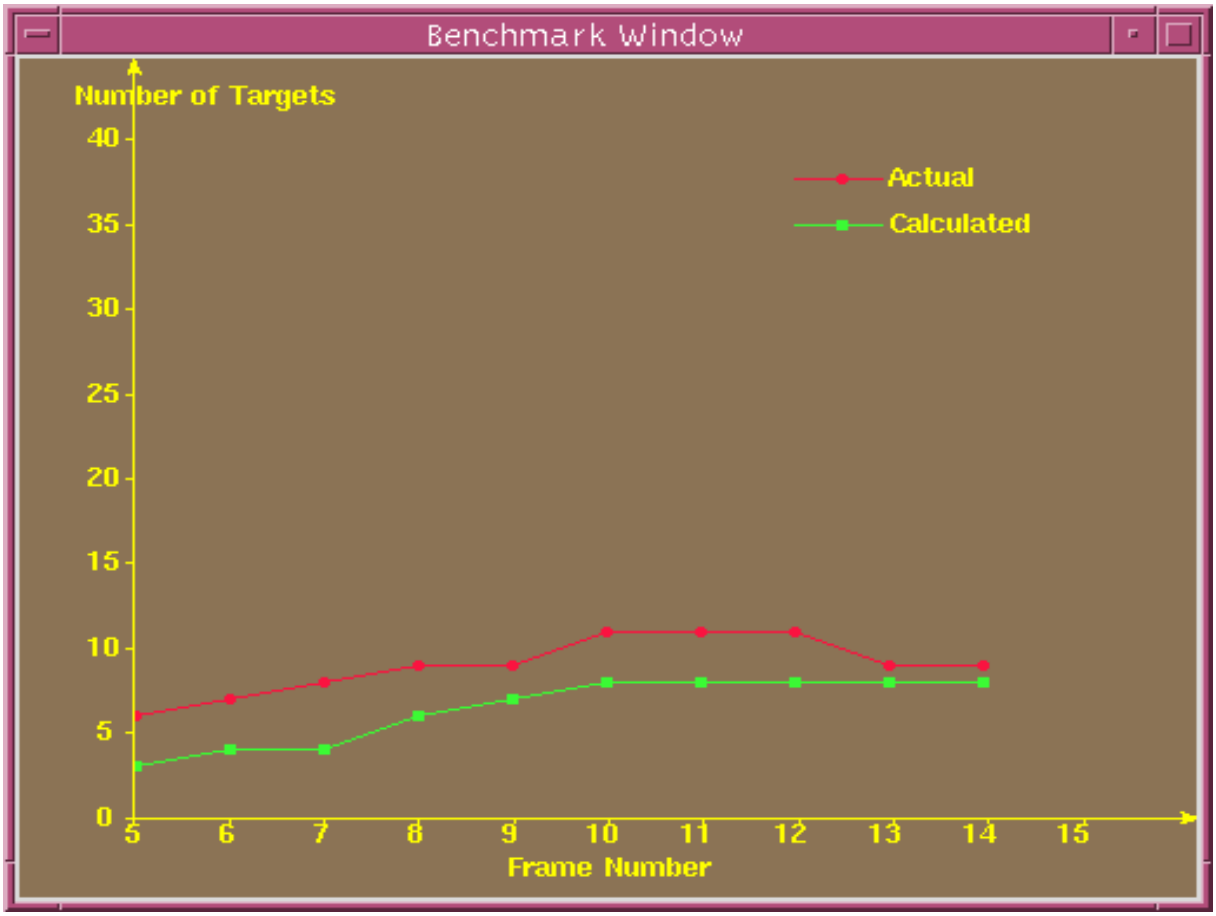
Multi-Hypothesis Tracking Benchmark

- A real-time target tracking benchmark of the DARPA benchmark suite

Implementing the Benchmark in RAPIDS

- The benchmark tasks are executed and results, in terms of targets sucessfully tracked, are returned


Benchmark Window
Number of Targets
40
35
30
25
20
15
10
5
0
Actual
Calculated
0
1
2
3
4
5
6
7
8
9
10
Frame Number


Benchmark Window
Number of Targets
40
35
30
25
20
15
10
5
0
Actual
Calculated
5
6
7
8
9
10
11
12
13
14
15
Frame Number

Results

- RAPIDS: A distributed simulator testbed for reliable real-time systems
- Helps designers compare different policies and algorithms for reliable real-time systems design
- Modular and easily extendable for new network protocols, allocation and scheduling algorithms and recovery techniques
- By exposing the potential bottlenecks, can help fine-tune the design of the architecture and operating system for a given real-time system

- **A preliminary version of the simulator has already been transferred to JPL and Cal Tech.**
- **We plan to have a student spend several months at JPL to tailor the simulator to their needs.**